

COS1004 Computer Systems Assignment 2 Part B

James Hassall, 102100517, COS10004

23 October 2019

Introduction

I am trying to make the classic Rock, Paper, Scissors (RPS) game in assembly. I initially tried to make a calculator in assembly but realise the time to implement such an application is too long. I decide to make RPS because I could make it relatively easy, where there are two inputs and 3 outputs (including the screen). The user makes a selection either Rock, Paper or no button down for Scissors, then the computer will make a decision after 4 seconds and display its answer. An indicator LED will light up red or green depending if the player lost or won respectively.

Design Outline

Physical components of the build was 2 10k ohm resistors, 2 1k Ohm resistors, 2 buttons, 2 LED's and a lot of wiring, the physical setup of the system is relatively easy due to the simplicity of the hardware. The software components was written in FASM because that is what I have been taught. The functions I have created are the 3 different actions Rock, Paper and Scissors where they link

```
;read first block of GPIOs
ldr r9,[r10,#52] ;read gpios 0-31
tst r9,#1024 ; use tst to check bit 10
bne Paper ;if ==0 e.g user selects Rock

;ldr r9,[r10,#52] ;read gpios 0-31

bl led1
bl setup_chars
adr r2,rock ; R2 = Text Offset "Scissors"
bl DrawChars
b cont

Paper:
tst r9,#2048
bne Scissors ;if user selects Paper
bl led2
bl setup_chars
adr r2,paper; R2 = Text Offset "Rock"
bl DrawChars
b cont

Scissors:
bl ledoff
bl setup_chars
adr r2,scissors ; R2 = Text Offset "Paper"
bl DrawChars
b cont
```

Figure 1: The main loop

up to a created function called DrawChars (located in DrawChar.asm) which will handle the drawing of the characters on to the display.

The program first waits for a user input and declares a response respectively, if the user selects scissors (e.g no input) the lights do not turn on and declare

a draw. When the user selects rock it will turn on the green LED and declare the user the winner, then if the user selects Paper it will display a red LED indicating the user has lost.

```
led1:  
    mov r8,#1  
    lsl r8,#18  
    str r8,[r10,#28]  
    bx lr  
  
led2:  
    mov r8,#1  
    lsl r8,#23  
    str r8,[r10,#28]  
    bx lr  
  
ledoff:  
    mov r8,#1  
    lsl r8,#18  
    str r8,[r10,#40]  
    mov r8,#1  
    lsl r8,#23  
    str r8,[r10,#40]  
    bx lr
```

Figure 2: The led functions

The other functions created are just initialising the GPIO pins and defining the implementation of the screen so the users input can view their choice visually.

Assumptions

I have assumed that using the GPIO reference doc that was in the earlier labs was fair to use.

Unresolved Problems

As of current build the AI only ever chooses Scissors and therefore the user can always win.

Running Program

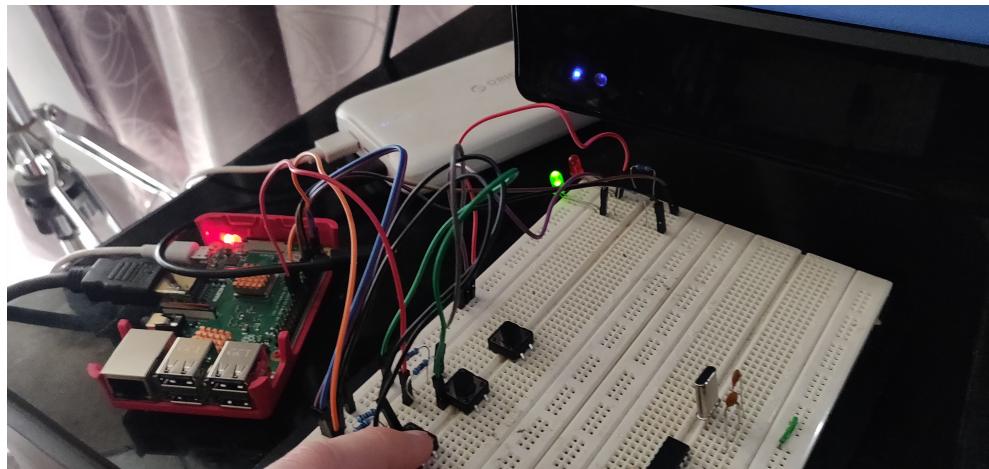


Figure 3: ledon

